Slicing Books - The Author's Perspective

Dahn, Ingo ; Armburster, Michael ; Furbach, Ulrich ; Schwabe, Gerhard

Slicing Books – The Authors' Perspective

Ingo Dahn, Michael Armbruster, Ulrich Furbach, Gerhard Schwabe

Abstract

The next generation of hypertext documents will be based on richer semantics and have more potential for automation and personalization than is currently possible. One system that shows much promise is the "sliced book". Slicing Book Technology splits textbooks into small, self–coherent slices and then uses semantic links to make structural relationships such as "Slice A builds on Slice B" explicit. This approach allows the reader to create personalized textbooks. The creation of sliced books poses new challenges to authors and "slicing book re-engineers". New tools support them during the creation of those features of electronic documents (multiple links, personalization) that are also specific for hypertext documents and are particularly relevant for learning. We therefore envision that learners will actively participate in the creation of sliced books. As a first step in this direction, this paper presents the technology and the experience of slicing book authors. The paper first introduces the Slicing Book Technology. Then the process of slicing books is described, presenting several approaches on how to create a sliced book. The fourth chapter contains a description and analysis of the experiences gained so far in the creation of sliced books. The final chapter describes the new learning approach we are currently working on: collaborative slicing.

1 Introduction

While authors are still struggling to understand how to make best use of the potential offered by hypertext documents, computer science researchers are already developing the next generation of digital documents. This next generation will be based on richer semantics, more potential for automation and personalization and will pose new challenges to the authors. This paper aims to give a first impression of these challenges. It is written by the developers of a new digital document type - called sliced books - and reports on their experiences during the first trials. Sliced books require a comprehensive support for writing semantically-linked documents. New tools support authors and slicing book re-engineers during the creation of those features of electronic documents (multiple links, personalization) that are also specific for hypertext documents and are particularly relevant for learning. We therefore envision that learners will actively participate in the creation of sliced books. As a first step in this direction, this paper presents the technology and the experience of slicing book authors.

The paper will first introduce the Slicing Book Technology. Then the process of slicing books is described, introducing several approaches on how to create a sliced book. The fourth chapter contains a description and analysis of the experiences in the creation of sliced books gained so far. The final chapter describes the new learning approach we are currently working on: collaborative slicing.

2 Slicing Book Technology

Hypertexts offer readers new qualities of service. Links take readers instantly from one point to any other point which is referenced to at this spot. Thus the classical hierarchical document structure is augmented or even replaced by a network of pages. However, still, information that is needed for a specific purpose remains scattered within this network. Slicing Book Technology goes one step further by compiling this information into a new document. Thus it can be seen as a combination of hypertexts with the dynamic generation of documents.

A sliced book consists of

- a hierarchy of semantic units together with

- a set of meta-data.

Unlike web pages, semantic units need not be complete documents that are ready for delivery. Rather they can be pieces of information that have to be combined with other pieces and processed appropriately in order to obtain a deliverable document. For example, to obtain a deliverable HTML page a number of fragments of HTML code may be combined into a new file, augmented with HTML header and footer and with an appropriate style sheet. This process may be varied in many ways. It may depend on characteristics of the particular user who has requested that page whether a certain fragment is included or not. Also the particular style sheet may be chosen to suit the users preferences.

This offers many possibilities for the production of personalised documents. In order to select among these possibilities, the aforementioned meta-data are used. These meta data contain for each of the semantic units information about this unit and about its relation with other units. An intelligent system needs that information in order to decide whether this unit should be included in a requested document. For example the meta-data for unit *1/1/3* may state that *it* is an exercise on topic *set theory* that requires the prerequisite units *1/0/5* and *1/0/10* for its solution. A teacher looking for exercises for an exam on set theory may just get *1/1/3* delivered but learners may also get *1/0/5* and *1/0/10* in order to prepare for that exam (Figure 1).

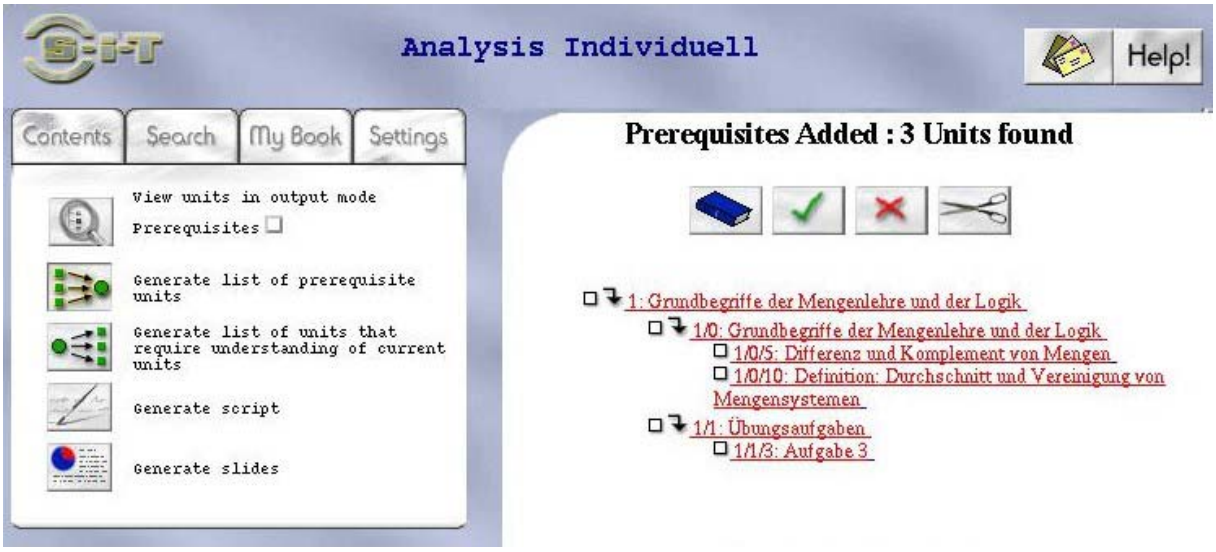

Figure 1: Example of a Sliced Book

From the point of view of the *reader*, the generation of a personalised book is done in two main steps.

1. Compose a table of contents that contains the selected slices
2. Request the generation and online delivery of the final document.

There are several tools available that assist the reader in the composition of a table of contents. The most familiar ones are the search functions. They yield a list of all slices that match the search criteria. Once on a screen, the reader may decide to put particular slices into his personal book. In this way, the reader can assemble any collection of slices in the table of contents of his personal book.

Similarly the button with the scissors on it may be used to remove slices from a given table. The buttons on the left of the screenshot in Figure 1 that generate the list of prerequisite units and the list of units that require understanding of the current units also produce appropriate

content tables. These functions make use of the semantic links between the slices that are stored in the database on the server.

Currently, the most advanced function is the script function. It generates a table of contents of all slices that lead the reader from slices he or she has marked to slices that constitute the current learning objectives without detours. This is just a first implementation of a powerful general principle.

The knowledge how to compose and format documents for teachers or learners is formulated in a general way, i.e. independent of the concrete units. Appropriate knowledge management procedures combine this general knowledge, the meta-data describing the content and the available information about the user and his current interest. They infer which slices should be presented to the user and how the document should be formatted. If possible, the knowledge management will resolve conflicts and select among possible alternatives.

When the reader wants to see the content, a click on the button with the magnifying glass in Figure 1 lets the server generate a new document consisting of the collected slices only. This document will be instantly delivered over the web in the same format in which the complete book would have been available for delivery, for example as a single pdf document. When new documents are combined from different sources, conflicting document styles may enforce the delivery of several separate documents that can be accessed through a common table of contents.

We note that the described architecture - consisting of a collection of meta-data annotated slices, a knowledge management system and a dynamic generation of personalised documents - coincides with the vision of the future World Wide Web as a semantic net. This vision was stated by Tim Barners-Lee – the inventor of the World Wide Web – at the conference XML 2000.

The first server using Slicing Book Technology has been online since May 2000. It can be accessed at http://www.slicing.de/books/ (Wolter & Dahn, 2000). The project "**T**ools for **R**eusable **I**ntegrated **A**daptable **L**earning – **S**ystems/standards for **O**pen **L**earning **U**sing **T**ested **I**nteroperable **O**bjects and **N**etworking" (Trial-Solution) started in February 2000 and investigates the possibilities of combining documents out of slices from various sources (see http://www.trial-solution.de). Six universities, two publishers and four other institutions co-operate in this project. It is funded in part by the European Commission within the 5th Framework program. In the project currently (December 2001) a repository of more than 25,000 slices from 3,600 pages obtained from 11 books is handled.

In order to make Slicing Book Technology work, collections of interoperable annotated slices must be available. Such collections can either be created from scratch or they can be generated by the adaptation of existing documents. Subsequently we shall concentrate on these authoring processes. Our report is based on the experience we gained during the authoring of the sliced book (Wolter & Dahn, 2000), during the preparation of further pre-existing books within the Trial-Solution project and during the writing of another book specifically for the use with Slicing Book Technology.

3 The Process of Slicing Books

3.1 Production of Sliced Books

The demand for high quality content, available online, is continuously increasing. This is an urgent issue, especially in the fields of online supported education and distant education. However, designing materials specifically for electronic media is currently rather expensive and it requires a combination of competencies that hardly any author can provide. On the other hand there is a large number of approved and appreciated documents available in print. In this section we will describe the process that makes these documents appropriate for online delivery, personalisation and added value electronic services. A description of the benefits that can be obtained in this way can be found in the reference (Dahn, 2000).

For using Slicing Book Technology, an existing document must be disaggregated into a set of slices and these slices must be augmented with meta-data. Since slices are intended for reuse, the content should be organised in a well-structured way so that the isolation of reusable units is possible. Well-written textbooks and legal and technical documents are good examples. The personalised documents that are generated will be composed of different sized slices.. The formatting may also be adapted to the users needs. Therefore, documents in a data format that intimately link their contents with their layout are hardly appropriate. We mention QuarkXPress and Postscript here. XML-based documents are ideal candidates since they are clearly structured according to a Document Type Definition and have the layout separated in their corresponding style sheets. LaTeX documents are also appropriate and Microsoft Word documents can be used.

The preparation of a document is performed in two large steps. In the first step, the document is automatically disaggregated and meta-data are automatically assigned. The second step consists of a manual revision of this automatically generated sliced document. Before a document can be sliced, the required granularity has to be decided upon. The size of a slice may vary between a chapter and a single line. The more finely grained the structure is, the more flexible possibilities for the generation of documents are obtained.

However a more fine-grained structure also results in a larger set of slices and requires more meta-data to be added for the description of these slices. Therefore the intended quality of service and also economic considerations determine which parts of a document are stored as separate slices. The following definition may help to determine appropriate parts (see (Dahn, 2001)).

Definition. A potential slice is a connected part of a document that can be reused under well defined conditions.

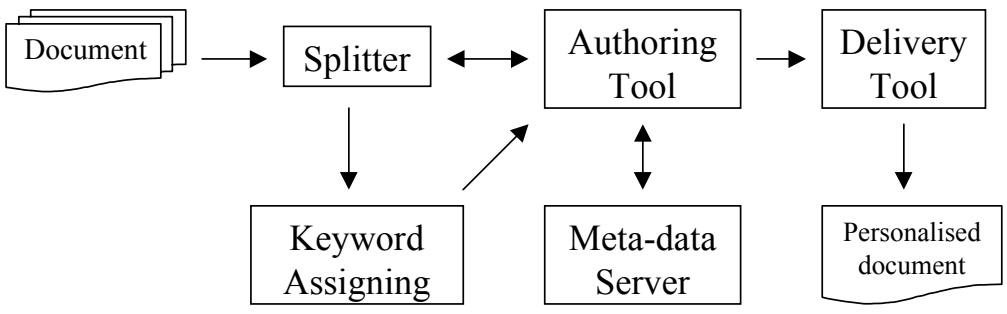

Figure 2: Tools used in the Process of Slicing

Splitting up a book into slices can be done by an automated tool, called the splitter, from the Slicing Information Technology, Berlin (SIT) (for the system architecture ref (Dahn & Schwabe, 2001), for the tools ref. Figure 2). Chapter and section headings are most easy to recognize automatically as borders of slices. However, figures, tables exercises and examples are also often clearly marked and can be extracted as slices. The way in which these elements are annotated varies from author to author. Therefore the existing automated slicing tools must be specifically configured for each author. A single configuration may serve them all only if a group of authors uses the same style.

The result of this slicing process can be conceptually interpreted as a tree of files. Leaves on this tree are atomic slices while inner nodes represent aggregations of slices. Each node may have a file with its specific content and two more files, named start file and end file. The purpose of these files is to ensure that generated documents are technically well formed. To achieve this, whenever the content of a node is to be included into a personalized document, this content will be enclosed between the start and end files of all parent nodes of the given node. For example, if the list

"A sliced book consists of
- a hierarchy of semantic units together with
- a set of meta-data."

is to be disaggregated, the list may be represented by a node that has a descendant for each item of the list and eventually the start and end tags of the list(<ul> and in case of an HTML document) in its respective start and end files. Furthermore the start file of the list will contain the introductory phrase "A sliced book consists of", which is valid for each item of the list. In this way, whenever the second item is to be presented, it will be enclosed between the appropriate list tags and the introductory phrase will always be presented, too:

"A sliced book consists of
- a set of meta-data."

The next step is the automated assignment of meta-data. Meta-data are all information that describe slices. Meta-data have to fulfill several needs. In order to retrieve the slices that satisfy a user's need, the content of the slices must be described in an appropriate form. The simplest way to do this is to use keywords. Perhaps keywords have already been assigned by the authors. Such keywords can easily be extracted automatically. In electronic publishing, keywords can be used much more flexibly than in printed books. They can be embedded into a thesaurus that also indicates relationships between keywords like synonyms or related concepts. These relationships can be taken into account when a reader starts a keyword search.

Unfortunately, authors do not define keywords in a standardized way. Therefore – in order to enable a uniform search for a library of sliced books – their keywords must be mapped into already existing thesauri. It is normal that authors define keywords that cannot be found in any available thesaurus. In these cases the new keywords must be standardized and incorporated using appropriate tools. A central meta-data server will be developed for this purpose within the Trial-Solution project

More keywords can be assigned by using automated indexing techniques. A number of such techniques have been developed, especially for the indexing of large sets of web pages. These techniques are based on statistical or on linguistic considerations. For fine-grained sliced

books, the atomic slices may contain too few data to apply automated indexing techniques. In these cases the indexing must be applied to larger aggregations of slices so that the slices can inherit the assigned keywords from these aggregations. The automatically assigned keywords must also be integrated into an existing keyword system.

Another type of meta-data describes the types of information that is contained in a slice. For example, a slice can contain an example or an exercise. These types can frequently be determined automatically either from the way in which these slices are formatted or from the position of the slice in the structure of the document.

More complex meta-data describe how slices are related to each other. It is most important to define which slices are pre-requisites for understanding a given slice. The automated extraction of hyperlinks from the source document can serve as a first draft. The analysis of phrases like "see also" or "using formula ..." can also provide hints. Usually this information is only in part explicitly available, but whenever it is found in the document it is used to add links between slices automatically. Of course these phrases also help the re-engineer of the automatically sliced document to understand the context better and manually add more meta-data later. But for the time being the splitter itself cannot "understand" the actual meaning of the document, so it is not able to extract all meta-data which relate slices to each other.

The meta-data described so far are the basis for determining which slices are to be presented to a reader in a concrete situation. Other meta-data can give the author and the publisher the opportunity to control how this is done. Most important is the inclusion of information that describes intellectual property rights. It must be possible to trace the authors, source and original location of each slice included in a personalized document. This kind of meta-data is usually inherited from the description of the complete document. The author or publisher may also wish to restrict the re-use of a set of slices. For example, he may not allow the rearrangement of slices or the combination with material of a competitor. This information is not contained in printed books and hence cannot be extracted automatically.

The meta-data system described here can be extended in various ways in order to provide additional services if this is useful. Some international consortia, for example IEEE, Dublin Core and IMS, have specified their own meta-data systems. None of these extensions is well-suited for automated meta-data assignment.

3.2 The Evolution of Re-Engineering and Authoring Approaches

During the development and tests of sliced books, we discovered the following approaches in the creation process: Slicing as re-engineering, slicing as authoring and slicing as iterative re-engineering and authoring.

Slicing as re-engineering: Initially, we took a straight-forward approach to the process of creating a sliced book: A slicing book re-engineer was given the digital version of a book. He first cut the book into slices and in a second step defined the logical connection between those slices. The resulting sliced book could then be used to create personalized documents (Figure 3, left side).

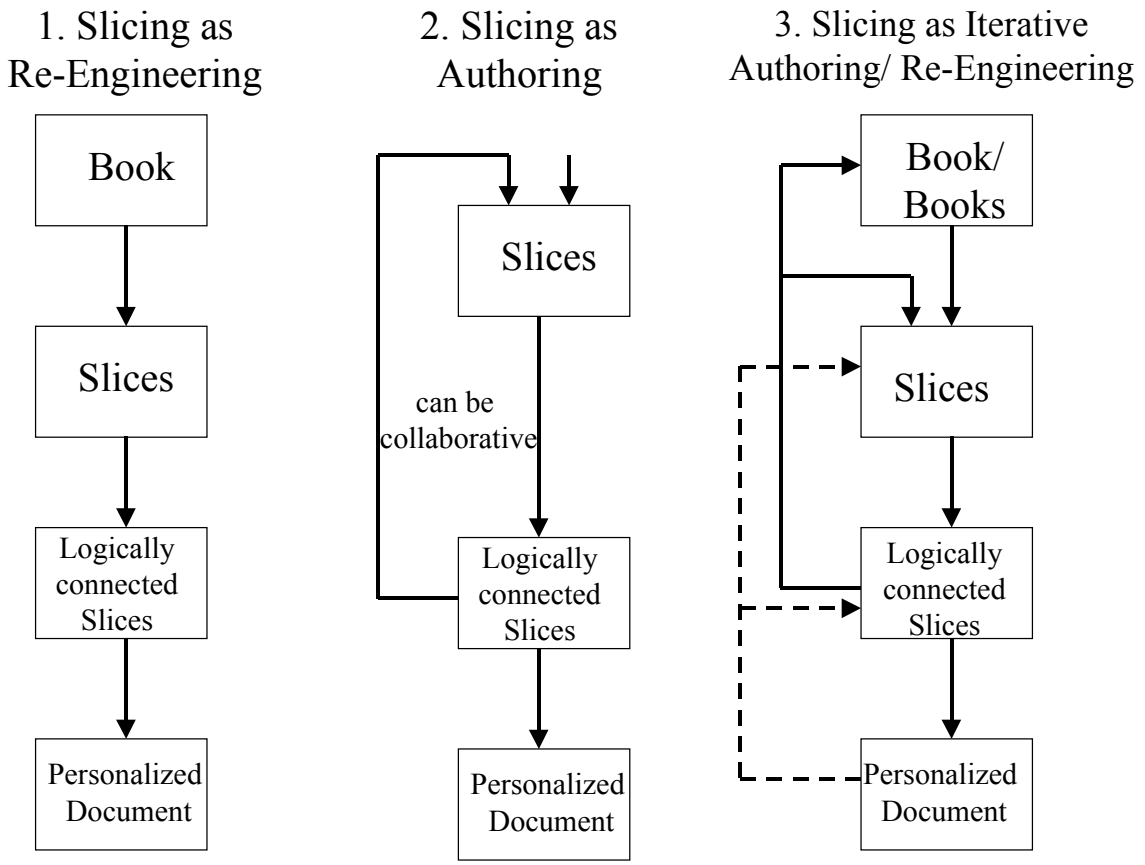

Figure 3: Approaches to Slicing Books

We call the person a re-engineer because he does not add any content (as an author does), but his task is more sophisticated than the work of a conventional book editor (who can abstract much more from the content he has to edit). The first section of Chapter 4 (Experiences) will reflect on the work of one slicing book re-engineer.

This re-engineering approach works well if
a) there is a complete book as a starting point
b) the objective is to produce a high-quality sliced version for a large number of consuming readers,
c) the slices can easily be identified by the re-engineer.

If the author himself is the re-engineer and is doing the slicing himself after the manuscript has been finished, he will have clear ideas about the constituents (slices) of his work and of their relationships. This was the case in the preparation of (Wolter & Dahn, 2000)[1]. If the re-engineer is a different person with competence in the respective domain, slicing becomes more of an iterative nature: The re-engineer proposes a set of slices to the author, discusses the proposal and problems with the author and then revises the slices. This is the typical situation found in the Trial-Solution project. The authors here have agreed to the re-engineering of their work and are ready to give advice, but they want to invest a rather limited effort.

[1] The author may even mark up his manuscript before the automatic slicing to make sure that it will be successful. Then relatively little manual restructuring will be required during the re-engineering phase.

Not only the collaboration between the author and the re-engineer turned from a sequential to an iterative process, but also the use of the automatic splitter tool and the manual re-engineering, bringing in another person which we call the slicing expert.

In our first attempts, the automated decomposition of the document was done as the first processing step as soon as the source was received from the author. The splitter slicing tool was used with default settings. It turned out that the slicing obtained in this way was often too fine-grained. The splitter, however, is adaptable to various levels of granularity and to a large variety of writing styles. We found it most convenient that the re-engineer, who is a domain expert, had a first look at the unsliced document from an end user's point of view and determined how it should be sliced. Then the slicing expert analyzed the document from a technical point of view and configured the splitter in an appropriate way. The resulting sliced document was imported into the authoring tool and inspected by the re-engineer and eventually by the author.

This process may have to be iterated several times. In this phase there are four ways to solve problems.

1. The slicing expert reconfigures the splitter.
2. The re-engineer or the author adds information (invisible in print) to the document to assist the automatic slicing process.
3. The slicing expert provides a special tool to add this information automatically.
4. The problem is deferred to the manual re-engineering phase.

In most cases the first of these variations is feasible. The others are required only in case of major irregularities in the document. The manual re-engineering starts only when the granularity, structure and meta-data are established..

Even then the slicing of some parts may have to be revised automatically during the re-engineering process. Then the re-engineering tool will export the current sliced document and the splitter will re-process parts, maintaining the results of the previous re-engineering work as much as possible.

Collaboration becomes more complicated when several re-engineers work concurrently on related parts of the same book or on related books. The main problem here is that one re-engineer has to set references into a structure that may be changed by a second re-engineer. Currently this requires considerate co-operation between the re-engineers. This can be supported if the related documents are considered as parts of a common super-document. For example, if a re-engineer sets a reference to a slice already handled by his partner, he will see the proposed key phrases that the partner had already assigned.

Slicing as Authoring: In the In2Math project, we are faced with the situation that a sliced document is produced from scratch, i.e. there is no book as input. In this situation, slicing has to be integrated with authoring. Slicing can become an integral part of the authoring process in a similar manner as the usage of the outlining tool of an editor can become an integral part of the writing process for a conventional electronic document. Slicing becomes authoring (Figure 3, center). The author first creates the slices and then incorporates them into the network of existing slices and relationships. The second section of the chapter on Experiences will reflect on the work of a slicing book author. The work is currently still seriously hampered by the lack of appropriate tools. While the role of the automatic splitter is reduced, the author needs integrated slicing and relationship-building features in the writing

environment. The product of slicing as authoring is again a sliced book that can be used to generate personalized books.

This authoring approach works well if
a) there is little or no material as a starting point
b) the objective is to produce a high-quality sliced version for a large number of consuming (!) readers,
c) the slicing process can easily be embedded into the writing process.

The chapter on Collaborative Slicing will describe scenarios and first experiences and how several actors can organize their collaboration in slicing books. If some of the collaborating actors are students, it may even turn out to be economically feasible to produce slicing books for a smaller number of active (!) readers.

Slicing as iterative re-engineering and authoring: The benefits of the sliced books should lead to improved traditional books, too (Figure 3, right side). The basic idea is to combine slices from several books into a slicing pool[2]. From this slicing pool, all authorized users can then generate personalized books across book boundaries. If a certain selection is very popular, the publisher can then use the selection as the basis for a new (conventional) book. We now envisage even more complex feedback mechanisms.

Reader feedback as well as new developments in the respective field may suggest a revision of the sliced documents (right part of Figure 3). This may affect the content of slices but also the assigned meta-data or rules for document generation. We have experienced cases where the suggestions of the intelligent advisory engine for the reader could be improved by correcting relationships between the key phrases following an analysis of user feedback. Note that this is very efficient since it will immediately improve recommendations for all books using this thesaurus.

For the future we envisage even more complex collaborative scenarios. Readers can compose their personal book from slices from different authors. This may put the work of the individual authors in a valuable new context. The resulting composed document will naturally be a new sliced book which can then be re-used by others. In this scenario a new role is created for the person who composes the document – not only as a reader but as an editor of a new sliced book. When this process is iterated, the editor will not only eventually collaborate with the authors but also with other editors.

This iterated process poses new questions concerning the author's rights. These questions have been discussed in the Trial-Solution project. We suggest that a reader can only forward the structure of a composed book for re-use; other readers will have to refer to the original sites of the copyright owners in order to retrieve the content. Every composed book will contain meta-data that allow the reader to trace each part of the document back to its very first source and author, also stating the person(s) who provided the compilation. The Trial-Solution meta data-system takes care of this in very much the same way as in conventional collections of articles.

[2] This approach is currently being tested in the In2Math project.

4 Experiences

After introducing the different approaches to the creation of sliced books in the previous chapter, this chapter will report on the experiences of a Slicing-Book-reengineer and of the a Slicing-Book author. Both are co-authors of this paper.

4.1 Experiences as a Slicing Book Re-Engineer

In this part of the article we would like to reflect on experiences gained with the manual revision of sliced books by the slicing book re-engineers at the Chemnitz University of Technology. The manual revision of sliced books includes the following operations which will be individually described below:
- checking of the slicing structure,
- assignment of titles, keywords and types to single slices,
- assignment of references between slices.

The work on the sliced version of "Mathematik - ein Lehr- und Übungsbuch" by Regina and Carsten Gellrich took several months. It is a textbook on mathematics for students of technical and economic sciences at technical high schools and colleges. The authors assumed that their readers do not want to deal with mathematics as a science but to use it for solving problems in their special fields. Consequently, no proofs and theoretical research are presented, but many examples, exercises and solutions were included. The book consists of 479 pages divided into eight chapters on the following topics: set theory, basic arithmetic operations, equation-solving, simple equations of higher degrees, linear and quadratic functions, proportions, powers, radicals and logarithms, trigonometry, functions, complex numbers, inequalities and mathematical basics of computer science.

The process of manual revision started on the sliced version of the book delivered by the splitter slicing software from Slicing Information Technology Berlin. The re-engineering tool used was developed with the first author at the AI-research group at the University of Koblenz-Landau. It presents itself in a conventional Web-Browser like Netscape or the Internet Explorer. With the re-engineering tool the slicing book re-engineer gets access to the file system consisting of several thousand files which make up the whole book. No other software is needed. Each file, i.e. each slice, of the book can be manipulated on its own.

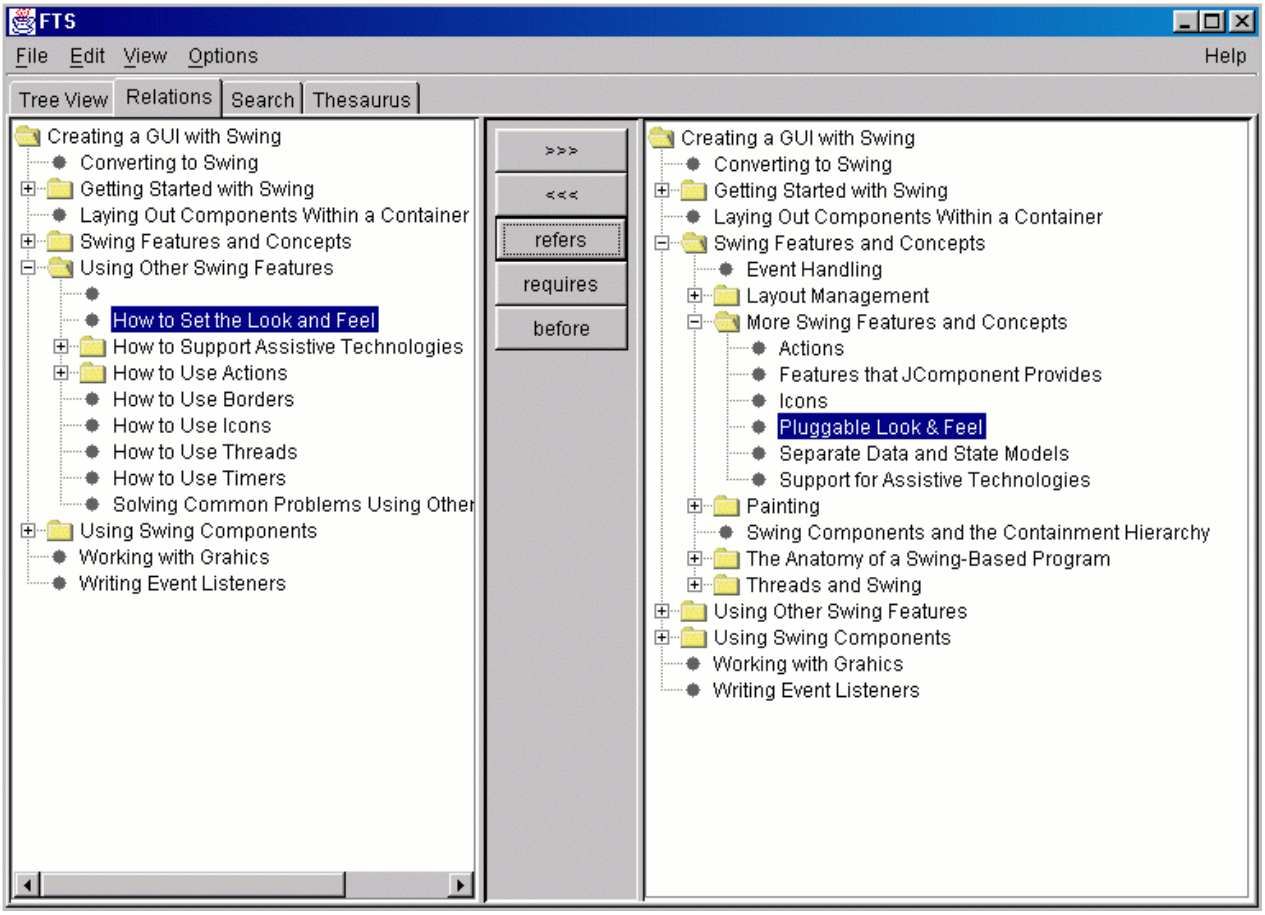

Figure 4: Screenshot of the Trial-Solution Re-engineering Tool

Now, what had to be done with the slices? First of all the structure of the slices created by the splitter had to be checked. The main question to answer for every single slice is: "Is the slice the correct size?"

A slice is considered to be a (basic) learning object, i.e., a bit of text which makes sense on its own and should not be split into smaller objects. In this context, for instance, a definition or a theorem, but also a single paragraph in an explanatory text, can be a (basic) learning object. In the cases of theorems and definitions, which usually use special environments in LaTeX, it is easy for both man and machine to extract them from the text and put them in slices of their own. It is hard to deal automatically with epic text, although the use of paragraphs by the original author might help the machine. So, bigger slices had to be split into smaller ones by reading and understanding the text and finding the points where the original authors changed from one learning object to another.

There were also slices which did not make sense on their own, for instance a split enumeration of the steps of solving a quadratic equation. One could argue that this could have been avoided by correctly slicing in the first place, but it is difficult to decide in advance where to split the text just by looking at the structures and environments of LaTeX and not at the actual occasions where these are used. Sometimes an enumeration cannot be split, on other occasions every item in an enumeration is a learning object of its own and should be separated from the other items.

An easier task to solve was the creation of titles for every small slice, although language consistency was an issue. Assigning keywords to every slice was more complicated. Keywords had already been assigned by the original authors, but it was obvious that the prospective user could ask for more which were not yet assigned and sometimes were even

not yet included in the used library of keywords. This means the slicing book re-engineer had to "make them up" using his knowledge of the subject. Still it was not easy to decide where to assign a keyword. Until now only human users of the sliced books and their demands were considered. It is well-known that a human looking for a special term with a search engine in the web can deal with a handful of found pages, but when the search results are several hundreds of pages he or she just cannot have a look at every single one of them. The same is true for a human searching in a book for a keyword which is assigned to too many pages or slices. On the other hand a computer can use thousands of these search results and deal with them in a sensible way. For the Slicing-Book-Re-Engineer it is difficult not to overshoot the limits of assignment of keywords usable by human readers and assigning the same keyword to too many slices although it makes sense at every single occasion. But on the other hand this might enable the user to a more specified search for several keywords in a slice.

Another problem is the assignment of types like definition, theorem, remark and exercise to every single slice. Often it is not easy to decide what the type of the slice in question is. Sometimes the original authors abused LaTeX environments defined for a semantic construct like a definition, to achieve a certain layout of the text though - from a semantic point of view – the content of the environment is no definition at all.

The main manual work on the sliced book is assigning references between slices. There are two different kinds of references. First of all there are references of the kind "The reader has to understand slice A in order to be able to understand slice B". For instance to understand a proof of a theorem the reader has to understand the theorem itself and all other theory needed in the proof like definitions, other theorems and so on. This kind of relationship between slice A and slice B is called a "refers" relationship. In slice B a reference to slice A has to be recorded. Once all references of this kind are set in all slices of the book the user can seek answers to questions like
- "What theory do I need to solve Exercise 135?",
- "What is the definition of a function used for?".
The other kind of relation between slices is the so-called "requires" relation between two slices. If slice B has a "requires"-relation set to slice A, then whenever slice B is presented to the reader so should slice A. This kind of relation can for instance be used when the text of a certain slice refers to a picture or table situated somewhere else in the book and therefore in a different slice.

To assign references to slices, and especially references of the first kind, it is vital that the slicing book re-engineer has a thorough understanding of the book's subject. Although he still has to read the book (and could thus learn the subject), this would prolong the time needed for the manual revision extensively. This is even truer when the book is large, deals with different subjects which are related or when a whole library of books with links between books shall be created. A slicing bookre-engineer who is not experienced in the subject could easily forget important references.

Most of the time needed for the manual revision of "Mathematik - ein Lehr- und Übungsbuch" was used to assign references to the hundreds of exercises in the book. There were similar exercises which basically needed the same theory to solve them, but the slicing of the theory in question was and should have been so detailed, that no single exercise could get assigned the same references as a similar exercise without solving both of them beforehand and comparing the solutions. For instance, there were exercises dealing with calculations of angles and sides of triangles. The theory includes the Sine-Theorem and the Cosine-Theorem. Which one can be used depends on the given values in the exercise. Of

course one could refer to both or to the whole theory before, just to be on the safe side, but this would not be satisfying for the potential reader of the book. Often several ways to a solution are possible, so all the theory needed for these different solution approaches should be referenced.. In this context the original author of the book can make life for the slicing book re-engineers much easier if he includes remarks (for instance on the necessary theory for solving exercises) in the source of the book when he writes it in the first place. Of course this is difficult if he also only takes some exercises from other material.

To conclude and look ahead to further developments, let us state the following: To the later user of the sliced book it might look like magic done by the computer, if he just has to press a button and the computer "explains" the relationships between different learning objects to him. But until now computers are not able to extract the needed references between learning objects unless the author has placed them explicitly in the text. So we still need manual revision and the search for techniques which help the slicing book re-engineers continues.

The activities of a slicing book re-engineer are highly relevant for learning: By making "builds-on" relationships between slices explicit, the re-engineer actively creates the overview knowledge so often lacking in conventional learning. Similarly, making relationships between descriptions and examples explicit gives the learner a deeper understanding of the text's rationale. Thus, fragments of knowledge do not only become connected in sliced books but also in the learner's mind. The networked knowledge structure of sliced books is not only more flexible than the sequential texts of textbooks, but gives also more structural clues than the usual links of traditional hypertexts. We will further elaborate on these opportunities in the section on collaborative slicing.

4.2 Experiences as a Slicing Book Author

One of the authors (Furbach) is currently working on a book on "Logics for Computer Scientists", which is intended to cover the material for a second year course in computer science. A LaTeX-document[3] is the basis for the book which was originally developed for transformation in HTML, so that the text could be used directly in the class room. This text does not include proofs of mathematical theorems (because these were presented in class on the blackboard) and it does not include motivational material (which instead was presented during the lectures). For the ongoing preparation of the book, all this has to be added to the already existing material. Altogether such a starting point appears to be very typical for book authors in an academic environment: there is material which was used during some courses and which has to be modified and extended towards a book. Typically an author wants to extend the material for the book in order to cover some more detailed and specialised aspects, which can be skipped during use in an introductory course. In our case the material now includes 60 pages together with links to interactive systems which are used in this course. It is currently extended and there are plans to connect it with other course material on similar topics.

Until now, the slicing book project has gathered experience from authors who delivered a finished text, which then was sliced automatically by the splitter. The author had to check and

[3] LaTeX-Documents make the formatting of documents explicit by embedding formatting instructions into an ASCII text. On the one hand this approach makes it impossible to create pure WYSIWYG-Editors. On the other hand, these formatting instructions allow much more sophisticated formatting than with editors like Microsoft work, e.g. the formatting of mathematical expressions. We used LaTeX documents for Sliced Books because most mathematical texts are edited in LaTeX and because the explicit formatting simplifies the re-engineering process.

update the slices (see the report given above); although this may have caused a lot of work, this method had the advantage of allowing the author to select an editing or desktop publishing system, as long as the result was a LaTeX-document. Hence, the author worked within his well-known environment.

The author has to use a special tool when the book has to be developed together with the slicing-structure and the meta-data. A prototype of an authoring tool developed by Ingo Dahn is available.

For authoring tools there are, in principle, two possible kinds of architectures:

- The authoring tool is responsible for bookkeeping and navigating through the slice-structure of the material. The author has to use this navigation tool to visit previously written slices and to restructure it. Whenever the author wants to modify or to add text, he browses through the structure to the location of interest. Then the authoring tool invokes a text editor which allows the input of LaTeX code for this particular unit.
- The author uses a text editor to browse through the LaTeX source code of his document, i.e. he can basically use his usual preferred method to write and manipulate the manuscript. He has to understand and modify the structuring commands which are included into the LaTeXsource if he wants to introduce new slices or to modify the slice-structure of the material, [4].

For the second method the author must be well aware of the data-structures which are used by the Slicing Book Technology. Of course, this can be hidden by incorporating these structure-browsing capabilities into the text editor, by modifying its implementation. Taking into account the fact that authors which produce LaTeX code usually are not using the de facto 'industrial standard' offered by Microsoft, we felt, that it would not be a good idea to force an author to use a specific editor. Hence we decided to focus on the first method, where it is no problem to invoke arbitrary text editors from the authoring tool.

Until now, the authoring tool was used more or less for the following tasks:

- Browsing and inspecting the structure of the document developed so far. This is like reading the book on a meta-level, which usually authors do not explicitly do. Suddenly the author finds himself in a situation where he is thinking how a future reader might access the document. There is a section in the introduction of most textbooks, which tells the reader what he is supposed to read in which sequence and which material he might skip through upon first reading. Usually this is depicted by a kind of dependency graph for sections. In our case this dependency graph is much more fine grained, its nodes are slices or units and, even more important, the graph cannot be defined a posterior, it has to be incrementally constructed during the writing process of the book. This structure definition has to be done by the author.
- Inspecting the content and the layout of the document or at least parts of the document. This task is even difficult in traditional writing because it forces the author to permanently switch between two different aspects: the layout of the document and the contents. Inspecting the contents without a minimal attention to layout and type setting is impossible because the LaTeX text formatting system will not produce readable output if the text contains typesetting and layout errors. In our case there is now a third aspect, namely the focus on a unit-based view, which forces the author to navigate

[4] Other similar approaches are possible once XML-editors become common.

through the text and to decide which units he want to preview as a document. The author has to navigate through the structure in order to read the contents.

- Preview of the entire or, at least, large parts of the document. For this task the authoring-tool was used to specify which units the author wants to preview, which then results in a display of the postscript file of the corresponding part. In this preview the units of the document are marked at the margins; these marks can be used to identify those parts of the documents which the author wants to work on.

Altogether we have three different aspects of a document – the contents, the layout and the slicing structure (including the meta-data).The author has to pay attention to all three aspects while developing the material. For each of these aspects we have tools to works with: the editor to produce the text, the LaTeX system for typesetting and the authoring tool for navigating. The more these systems are separated from each other, the more difficult the task is for the author. He has to do three jobs at a time: writer, designer and knowledge engineer.
Let's take, e.g., the task of extending a subsection within the text. The author first has to localise the appropriate unit: for this he browses with the re-engineering tool (Figure 4) through the slicing-structure. In order to check the contents of a unit, he invokes the preview-option of the tool and reads the content of a unit and its immediate context. When he finds the appropriate unit for the planned extension, he invokes the text editor for this unit and is now able to introduce new text. However, the editor only contains the unit he is working on together with some immediate context; in order to check some cross-dependencies the author has to again enter the unit-browser, navigate to the unit in question and again invoke the preview-options. During all this, the usual layout-tasks have to be solved by including LaTeX commands in the text. Finally the meta-data for the extended unit, like keywords and dependency descriptions, have to be inserted. Altogether, the author has to change the tools and the view to his book several times throughout this sample-task; he constantly has to switch between the different views of his document.

On possibility for facilitating the above task, would be to integrate the browsing tools for navigating through the slicing-unit structure into the text editor or the layout-preview tool.

An integration into the text editor would certainly reduce the author's workload. He is used to producing text together with layout LaTeX commands by means of his editor; the additional work in taking care of the slicing structure seems to be manageable. As mentioned before, the disadvantage of this solution might be that the editor is fixed. The alternative solution, namely to present the structure together with the preview of the documents, would certainly be appropriate if the preview allows interaction. In this way, the author could use the layout-preview to browse through his document and to invoke an editor from there.

Figure 5 shows an example of such a combined previewing-browsing approach in the recently introduced version of the German newspaper *Rheinzeitung* (see http://epaper.rhein-zeitung.de/01/05/30/). The user gets a preview of the newspaper and this graphical presentation can be used to browse through the newspaper, very much like the paper version.

Figure 5: Previewing-Browsing the Rheinzeitung

Parts of the presented pages are links which connect readers to the respective articles. If the user activates a link he gets the contents plus the graphical browsing possibility that is maintained via thumbnails of the pages.

4.3 Analysis of Experiences

The experiences we have collected permit us to derive some conclusions on the creation of sliced books. Note that we do not consider the cases where authors program complex multimedia applications. Please also note that we report on work in progress.

1. The process of creating sliced books is knowledge work.
It can therefore not only be applied to the production of electronic books by professional slicing book engineers and authors, but also as a means for learning. One of the more interesting uses involves the collaborative creation of sliced books by teachers and their students (we will elaborate on this scenario in the next chapter). A general observation appears to be that the slicing book knowledge representation can be created and used in diverse contexts without changes. But each context needs special functionality and often even a special tool to access and manipulate the knowledge structure. The reported experiences of the first slicing bBook author shows that the cognitive load is much too high if tools developed for one context are applied in another context.

Knowledge work can rarely be done in isolation, but builds on other information. For example, the document should have numerous key phrases assigned and each key phrase should be assigned to all parts that consider the corresponding topic as a major concept, not only to the first place where this concept is introduced. It is recommended that authors re-use key phrases that have already been assigned in related books. This facilitates the development of unified access tools. A detailed thesaurus of recommended key phrases maintained by competent personnel and available over the Internet would be a great help. It is intended that the thesaurus developed in the Trial-Solution project will be made accessible to the public through Fachinformationszentrum Mathematik/Informatik Karlsruhe, Germany.

2. Sliced books are constructed.
There are different approaches to writing texts or hypertext documents. The slicing book approach has a bias towards viewing document creation as a construction activity. The definition of modules and their relationship, typical for engineering, is an important part of the creation process. Much like an engineer, the creators of Sliced Books can build libraries of components and reuse them in different contexts. If compared to software construction, the creation of a sliced book bears the closest resemblance to declarative programming (e.g. with Prolog).

A particular future challenge will be to support the early phases of the creation of sliced books. Much as an architect needs a manual sketch before he plans a house in detail, the sliced bBook authors need semiformal representations of slices and their relationships before they define them in detail. A candidate for supporting early phases will be Mind Maps.

3. The creation of sliced books is not the same as the creation of classical hypertext documents, but there are some important similarities.

3a) Sliced books and guided tours: Slicing Book Technology enables the adaptation of documents for specific purposes. One of these is to serve as an introduction to a book or topic similar to a guided tour in hypertext documents. Software prefabricated books play this role in the existing SIT-Reader. These are recommended first readings for certain user groups (stereotypes, cf. (Rich, 1989)). Unlike a guided tour, a prefabricated book is copied to the reader's personal book on the server. He may read it, but more importantly, he can at any point augment it automatically with more information he is interested in. Therefore it is important to select such slices for prefabricated books that have numerous links to various parts.

Prefabricated books can be considered as initial user profiles recommended for certain groups of readers. Thus they can also recommend settings for electronic search procedures. For example the possibility to search for detailed proofs will not be selected in a prefabricated mathematical book that is designed for people who want to get a brief overview only. Prefabricated books are mainly intended for first time readers. They draw the attention of these readers to the parts of the book that are of special interest to them.

3b) Hyperlinks and relations: Hyperlinks take readers of a hypertext from one place of the text to another or to another document. Similarly, the *refers* and *requires* relation relate different parts of a document. However there are a number of important differences too.

- A hyperlink has no specific semantic; the relationship between slices typically has a semantic. This semantic richness changes the process of creating a sliced book (we would argue that it makes it easier for authors and slicing book re-engineers) and is the basis for automated support. The lack of semantic richness of hyperlinks is the basic argument for Tim Barners-Lee to propose a new Semantic Web architecture (Barners-Lee, Hendler & Lassila, 2001). The slicing book architecture contains a few predefined relationships, but the tools we currently use are flexible enough to support new types of binary relations between slices when they are needed by authors.
- Not all required relationships have to be explicitly represented. It is possible to introduce derived relationships by defining them formally on the basis of previously introduced relations and attributes. The automated inference systems that are used to compose personalised documents can resolve these definitions at runtime. This significantly reduces the work for the creators of sliced books: The numbers of manually defined relationships is typically lower than the number of hypertext links.

The author can rely on a few general definitions instead of defining many specific relationships. .

- Relationships connect (groups of) slices. Unlike positions connected by hyperlinks, these slices carry specific content. Thus relationships between slices can much better reflect relationships between contents.

Of course, ordinary hyperlinks can be used in addition to meta-data. Even if the text to be sliced was a hypertext, they can automatically be analysed in order to derive useful meta-data.

3c) Formatting: Hyperlinks are visible in the hypertext. They are intended for use by humans and need an explanatory text or image. In contrast, the reader of a sliced book will not see the meta-data. These are only intended for use by the advisory engine that helps the reader to collect the slices he is interested in. On the one hand this means more freedom to the author: He is free to add as much meta-data as necessary in order re-use the material. On the other hand the author has more responsibilities to ensure quality; while a hypertext reader may discover meaningless or faulty links by himself, he will quickly get lost if there are faults in the knowledge structure underlying a sliced book. If, however, slices are created and linked carefully, the reader will not as easily "get lost in Hyperspace" as with hypertext documents.

The author of a hypertext has much more control of the formatting than the author of a sliced book. In practice, the final layout of an HTML document will be not determined by the author but by the reader's Web browser. The author of a sliced book, however, cannot even be sure about the context in which his text will be re-used. The slicing book author will also design a presentation, but just as a kind of "master document". He should strictly separate the presentation from the content. The tools that produce the final personalised documents will try to respect the design of the author as much as they can but they will also change it where necessary – either to meet the requirements of the reader or to support the combination with content that had been formatted in another way. In cases where automated re-formatting is problematic, the author may additionally decide to provide alternative versions of some slices. The author can determine the conditions of re-use for each of these versions by using appropriate meta-data.

4. The creation of slicing books iterates between automated phases and manual phases. It is neither advisable nor feasible to automate the creation process (even if a completed book serves as an input), nor is it advisable or feasible to create a sliced book completely manually. This observation is interesting because the explicit representation of content knowledge significantly enhances the potential for automation in comparison to traditional hypertext documents. Still we find that all automated phases need some manual preparation or rework.

Given the sheer number of slices (1500-3000 in a fine sliced average book) it is essential to automate the slicing and meta data assignment as much as possible, knowing that perfection cannot be achieved automatically. In many cases meta-data are present in the documents, though they are not explicitly annotated. For example, we have sliced a collection of legal documents which contained laws, cases, addresses and comments. These different types of slices had not been mentioned in the text but they could be extracted automatically from the table of contents and inherited by the individual slices. The way in which meta-data are encoded varies greatly from author to author, which means the tuning of the splitting and meta data assignment tools is essential. The invention of meta-data pattern by which meta-data can be automatically assigned to groups of slices has considerably simplified the work of the re-engineer. These patterns are defined by the re-engineer himself. It is possible to propose pattern automatically using machine learning techniques.

Automated key phrase assignment is inherently difficult. Better linguistic tools could be an important help here but even then the linguistic material in a slice is frequently too small for a reliable semantic analysis. In these cases key phrase assignment must recur to the analysis of groups of slices. When the analysis has identified key phrases that describe concepts handled in a slice, it is necessary to distinguish between concepts that are indeed discussed in that part and those that are only used as auxiliary concepts. The current limited experience suggests that automatically assigned key phrases must be critically inspected by the re-engineer. Nevertheless automated key phrase assignment is very important since it suggests key phrases from a controlled vocabulary to the re-engineer. This helps maintain a coherent pool of key phrases.

Automation of the slicing process can be considerably improved by following some simple rules during the writing of the document. Above all, the document should be clearly structured. Start and end of re-usable parts should be clearly mentioned. This can be done in different ways for parts of different characters; e.g., examples can be annotated differently than exercises. This should be done in the same way for each type throughout the document. Style templates should be used instead of manual formatting whenever possible. In general, manual formatting should be avoided. Electronic re-use is not tied to the size of a print page and necessary re-formatting cannot be foreseen in advance. Therefore this is best left to the electronic formatting systems, web browsers etc.

5. Seamless integration into standard tools and the reliance on open document standards is paramount.
However different the creation of sliced books may be, many individual steps are derived from current work practices. We are already used to creating paragraphs of text documents and to format documents with standard word processors (such as LaTeX or Microsoft Word). Standard tools support even more sophisticated editorial activities such as the creation of keyword lists. The learning load is acceptable only if the user can apply his conventional tools to these slicing book activities. Standardisation also applies to the document formats. Currently there are numerous discussions on open standards for the description of meta-data for documents and learning objects. Choosing between these proposed standards is a hard task for an author. In our view this is a secondary question that should not concern the author too much. It is, however, important that the document contains a maximum amount ofinformation that can be used to recommend parts to an interested reader. This information should be included in the document in a uniform way. Then it can be later automatically extracted when needed. The upcoming XML-Standard encourages the production of structured documents and thus reduces the re-engineering work.

There is still considerable work to be done on developing better support for authors. Document structuring and the addition of meta-data should be supported by popular editors. Their conversion into DTD-based comfortable XML editors would be a great step forward in this direction. Separate tools for slicing and re-engineering are still required if other editors are used.

6. Sliced books offer authors the opportunity for new business models:
The online component of a sliced book can be easily changed. Outdated material can be replaced or augmented. Thus the author can modify his work based on feedback from the reader. When building prefabricated books, the author may realise that some parts are missing to make the content intelligible for certain outside readers. Then such parts may be augmented later or they may be pulled in from other sliced books. Alternatively, other authors may provide just the small parts that are missing in order to provide a link with their own material.

As with conventional citations, authors will have the right to set references to any slice from another author but they cannot be sure that links to their work are included in other works. When the author of a book sets a link to a book from another author, this does not imply that the readers of the first book have the right to access content from the second. For such situations, authors should prepare attractive information that they are willing to deliver for free instead of the content (for example an abstract of the content).

5 Collaborative Slicing

Up to now this text focussed on the creation of sliced books by individuals. The complexity of the creation of sliced books suggests making the creation of sliced books a collaborative effort. As we are currently only exploring how to proceed in this area, this chapter will only introduce scenarios for collaborative slicing.

There are at least three scenarios for collaborative authoring:
1. Author-Author collaboration: A group of authors and editors prepare a sliced book that will later be used by the reader.
2. Author-Reader collaboration: Authors and readers collectively develop a sliced book starting from a base manuscript. For example, a university lecturer and his or her student jointly develop a sliced book starting with the traditional lecture notes. Slicing a book is regarded here as a learning methodology.
3. Author-Others collaboration: After preparing the content of a book authors collaborate with other persons such as publishers, meta-data authorities and re-engineering staff in order to fine-tune and manage the content.

Ad 1: Author-Author collaboration: Collaborative slicing can be seen as a special case of collaborative writing. Collaborative writing is difficult because authors on the one hand need to be aware of what the others are doing, in order to produce a coherent result (Schwabe, 1995). On the other hand, they need times of undisturbed work in order to formulate their own thoughts. Typical tools for collaborative writing therefore distinguish between divergent (loosely coupled, often asynchronous) modes, where each author works on his own section of work, convergent (closely coupled and synchronous) modes, where all authors see the same content, and mixed modes where some authors work in a convergent mode and others in a divergent mode. Divergent phases further the deliberation of the content and ensure that all needed information is included in the product; convergent phases ensure that the product becomes a coherent whole (for a general introduction to collaborative tools ref. (Holmer, Haake & Streitz 2001), for collaborative hypertext ref. (Mark, Haake & Streitz, 1997), for a theoretical discussion on convergent and divergent phases ref. (Dennis & Valacich, 1999), (Schwabe, 2001)).

Collaborative slicing needs both convergent and divergent modes, too, but the activities in these modes differ from conventional tools. Although tools have been developed to reserve sections of any granularity for personal work, experiences show, that for divergent phases authors mostly reserve paragraphs or sections for their own work. In sliced books, individual authors will be inclined to reserve individual slices for their personal work; these may be, but do not have to be, equivalent to paragraphs or sections. Furthermore, sections can be recursively aggregated, slices cannot be.

The interesting activities, however, happen through the convergent phases. Here, the authors jointly prepare the meta-data. An easy approach could be to let one author do the physical editing and others just participate orally. This may not be very productive, so at a minimum, the passive authors need a pointing device to contribute or there needs to be support for taking

turns in active editing. A much more interesting approach might be a structured division of labour: Slices have (by definition) a module-like character. Thus, in a first step, each author may define pre-conditions and post-conditions for his slices. In a second step, the authors searches for other slices that have post-conditions that fulfil the preconditions of his slices. Here, both – a shared framework of reference and intelligent tool support - may be needed to make this feasible. If there is a slice that at least partially fulfils his requirements, he negotiates the interface with the other author as a third step. On the basis of this negotiation, both authors may also agree that it is necessary to rewrite the slice. If no existing slices fulfil his requirements, the precondition is noted on a list that is the basis for further writing. In a fourth step, the total structure of the document is discussed and settled in a group session with all authors.

These four steps can be iterated several times. Besides sufficient tool support, it requires good moderation skills within the group of the authors. We have not yet tested this approach, but it is useful to demonstrate how the modular and logical structure of sliced books can lead to new collaborative authoring approaches. Other approaches can be based on the similarity of slicing authorship to programming. Here one could try to transfer collaborative programming approaches (e.g. pair programming or approaches for shared program repositories) to collaborative slicing book authorship.

ad 2.: Author-Reader collaboration: In learning contexts, authors and readers can together construct their shared understanding in a collaborative writing scenario. Here computer supported scenarios can be particularly useful ((Bromme & Stahl, 1998), (Briggs & Brown, 1997)). First evidence from usage of our sliced book in University contexts indicates that authoring slicing text can be an even better learning instrument, because the relationship of knowledge pieces (= slices) are made explicit. Students switch between content understanding/preparation and reflections on the content. What could be a suitable scenario for such an author-reader collaboration?

The scenario can again be based on divergent and convergent modes: The teacher-author could prepare and present the content as unconnected slices in a divergent mode to a class of students. The students then split in subgroups of two or three persons and try to build the slicing relationships and thus explore the deeper structure of the content presented. In a third convergent phase the whole class tries to arrive at a shared understanding of the slicing structure. There can be several variations of this approach: a) the students prepare the slicing relationships at home and only the subsequent third phase is done in class. b) the author and the readers discuss the slicing relationships in an oral discussion. Here content discussion and meta-discussion can be interwoven. c) the author presents only parts of the necessary information. The readers then have to collect the other data and prepare the slicing hierarchy. This approach can be useful for case studies or experiments. d) besides the shared representation of slicing book content and slicing relationships exist individual annotations and individual slicing relationships. This approach would take individual preferences on knowledge representation into account. Current tools do not sufficiently support author-reader collaboration yet. Requirements for a future collaborative tool include

a) an easy to use interface for ad-hoc in-class slicing
b) shared access to a shared slicing representation (incl. concurrent access, locking, view linking, tele-pointing etc.)
c) personal annotations to a slicing representation
d) author-control of the collaborative process
e) functionalities to test the slicing representation

Currently we are working on building special tool for these purposes and have been involved in first exploratory experiments (Valerius, Dahn & Schwabe, 2001).

ad 3) Author-Others-Collaboration:
Sliced books gain considerable added value if their slices can be reused in combination with slices from other documents. In order to support this, the meta-data used for the description of the slices of a book must be integrated with the meta-data used for other books. This integration should be taken into account at an early stage of the authoring work. We envisage a central meta-data server from which authors can download already existing meta-data. This is necessary in order to encourage the author to use keywords that are also used in other books. Nevertheless, it cannot be expected that already existing meta-data are sufficiently detailed to describe specific finely- grained sliced books. Therefore, authors must have the possibility to define their own meta-data assignments.

Authors are domain experts. It cannot be expected that they will define their new meta-data according to some general standards. Therefore, they will have to co-operate with meta-data experts that can integrate the new meta-data consistently into existing systems. The author may also have to co-operate with experts that develop specific styles for the presentation of his book. Since slices can be easily linked to slices in other books or to other resources on the Internet, co-operation with authors of other books in order to settle issues of combination of slices is relevant. These issues also concern respecting intellectual property rights – an item that is also of importance for publishers.

6 References

Barners-Lee, J., Hendler, J. & Lassila, O. (2001)	The Semantic Web. Scientific American, May
Bromme, R. & Stahl, E. (1998)	Räumliche Metaphern und das Schreiben von Hypertext – Eine Studie im Schulunterricht [Room Metaphors and the Writing of Hypertext]. Zeitschrift für Pädagogische Psychologie, 12, pp. 156-166.2/3.
Briggs, R. & Brown, H. (1997)	From the Sage-on-the-Stage to the Guide-on-the-Side: Re-engaging the Disengaged Learner with Collaborative Technology, University of Arizona, Tucson. http://www.cmi.arizona.edu/personal/bbriggs/Downloads/aincost1.doc
Dahn, I. (2000)	Symbiose von Buch und Internet [Symbiosis of Book and Internet]. In U. Becker & W. Sommer (Eds.), Learntec 2000, (pp. 551 – 558). Karlsruhe: Karlsruher Kongreß- und Ausstellungs-GmbH
Dahn, I. (2001)	Slicing Book Technology - Providing Online Support for Textbooks. In Proc. 20th World Conference on Open Learning and Distance Education. Düsseldorf: Proceedings on CD-ROM
Dahn, I. & Schwabe, G. (2001)	Personalizing Textbooks with Slicing Technologies – Concept, Tools, Architecture, Collaborative Use. In Proc. 35th Annual Hawaii International Conference on System Sciences. IEEE Computer Society http://www.hicss.hawaii.edu/digilib.htm
Dennis, A. & Valacich, J. (1999)	Rethinking Media Richness: Towards a Theory of Media Synchronicity. In Proc. 32nd Annual Hawaii International Conference on System Sciences. http://www.hicss.hawaii.edu/digilib.htm
Holmer, T., Haake, J. &	Kollaborationsorientierte synchrone Werkzeuge. [Collaboration

Streitz, N. (2001)	Oriented synchronous tools] In: Schwabe, G.; Streitz, N.; Unland, R. (Eds.), CSCW-Kompendium – Lehr- und Handbuch zur computerunterstützten Gruppenarbeit (pp. 180-193). Heidelberg: Springer
Mark, G., Haake, J., Streitz, N. (1997)	Hypermedia Use in group Work: Changing the product, Process, and strategy. Computer-Supported Cooperative Work: The Journal of Collaborative Computing, 6, (pp. 327-368).
Rich, E. (1989)	Stereotypes and user modeling. In Kobsa, A. & Wahlster, W. (Eds.), Usermodels in dialog systems (pp. 35-51). Berlin: Springer
Schwabe, G. (1995)	Objekte der Gruppenarbeit - ein Konzept für das Computer Aided Team [Objects of Group Work]., Wiesbaden: Gabler
Schwabe, G. (2001)	Mediensynchronizität - Theorie und Anwendung bei Gruppenarbeit und Lernen [Media Synchronicity]. In Hesse, F. & Friedrich, H (Eds.), Partizipation und Interaktion im virtuellen Seminar (pp. 111-134). Münster Waxmann
Valerius, M., Dahn, I. & Schwabe, G. (2001)	Adaptive Bücher für das kooperative Lernen, Anwendungen – Konzepte – Erfahrungen [Adaptive Books for Cooperative Learning]. In Engelien, M. & Homann, J. (Eds.), Virtuelle Organisationen und Neue Medien 2001, (pp. 391-413). Lohmar: Josef-Eul-Verlag
Wolter, H. & Dahn, I. (2000)	Analysis Individuell [Personalized Calculus], Heidelberg: Springer-Verlag